L Number	Hits	Search Text	DB	Time stamp
1	550582	(detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:22
2	22633	(repair$4 or fix$4 or heal$4) adj3 (program$ or list$ or function$ or subroutine$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:30
3	155	((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$ or function$ or subroutine$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:22
4	221134	(detect$4 or track$4 or monitor$4 or check$4) adj5 (fail$4 or error$4 or problem$ or fault$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:23
5	23958	(repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:44
6	67	((repair$4 or fix$4 or heal$4) adj3 (program$ or list$ or function$ or subroutine$)) with ((detect$4 or track$4 or monitor$4 or check$4) adj5 (fail$4 or error$4 or problem$ or fault$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:24
7	8	(((repair$4 or fix$4 or heal$4) adj3 (program$ or list$ or function$ or subroutine$)) with ((detect$4 or track$4 or monitor$4 or check$4) adj5 (fail$4 or error$4 or problem$ or fault$4))) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:29
8	556967	remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:29
9	8215	(repair$4 or fix$4 or heal$4) adj3 (program$ or list$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:58

10	0	((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) with remote with ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:36
11	0	((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) same ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) same remote same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:36
12	2376	((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:36
14	1	((((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))) and remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:37
13	11	(((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:37
15	1		USPAT	2003/08/07 14:41
16	1		USPAT	2003/08/07 14:41
17	515	(repair$4 or fix$4 or heal$4) adj3 malfunction	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:49
18	24412	((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:50
19	2382	(((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:50

20	11	((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:50
21	0	(((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$))) not ((((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4)) with ((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:51
22	18	((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) and ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$)) and remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:51
23	17	(((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) and ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$)) and remote) not (((((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction)) with ((detect$4 or track$4 or monitor$4 or check$4) with (fail$4 or error$4 or problem$ or fault$4))) same ((repair$4 or fix$4 or heal$4) adj3 (program$ or list$)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:55
24	3370	(714/?).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:55
25	5007	(717/1??).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:55

26	0	(717/2??).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:55
27	0	(717/3??).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:56
28	0	(717/?).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:57
29	8352	((714/?).ccls.) or ((717/1??).ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:57
30	16062	(repair$4 or fix$4 or heal$4) adj3 (program$ or application$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 14:59
31	121	((repair$4 or fix$4 or heal$4) adj3 (program$ or application$)) with (((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 15:01
32	3	(((repair$4 or fix$4 or heal$4) adj3 (program$ or application$)) with (((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction))) same remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 15:03
33	34	(((repair$4 or fix$4 or heal$4) adj3 (program$ or application$)) with (((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction))) and remote	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 15:03
34	7	((((repair$4 or fix$4 or heal$4) adj3 (program$ or application$)) with (((repair$4 or fix$4 or heal$4) adj3 (fail$4 or error$4 or problem$ or fault$4)) or ((repair$4 or fix$4 or heal$4) adj3 malfunction))) and remote) and (((714/?).ccls.) or ((717/1??).ccls.))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/07 15:03

---------- KWIC ---------

Detailed Description Text - DETX (23):
   Once the network shown in FIG. 1C is established, Pete's Catering
computer
system can automatically **detect problems** with it's components
(hardware,
software, or microcode), build a service request describing the
problem, select
a SP/R responsible for **fixing the problem,** (either the Hardware Fixit
Shoppe or
the Software Fixit Shoppe) and send the service request to that SP/R.
The SP/R
responsible for **fixing the problem** receives the service request,
verifies that
Pete's Catering is entitled to receive service, and **checks** a solution
log to
see if it has the solution for the **problem**. If so, solution
information
describing the fix for the problem, accompanied by one or more software
components, microcode components, hardware parts order, and/or textual
instructions, is sent down to the SR. If the responsible SP or SP/R
cannot **fix**
the problem, it checks to see if it receives support for this **problem**
from any
other SPs or SP/Rs to which it is connected. If so, it sends the
service
request on to this SP or SP/R. This process continues until a fix for
the
problem is found.

Detailed Description Text - DETX (28):
   The computer systems in the service network of the invention also
have the
capability to perform or request problem prevention. A SP (or SP/R
acting as a
SP) can check to see if it has any solutions to problems that one or
more of
the SRs (or SP/R acting as an SR) it supports has but has not
discovered or
reported yet. If so, it can distribute solution information,
accompanied by
one or more software components, microcode components, hardware parts
order,
and/or textual instructions, to the SRs. In addition, a SR (or SP/R

acting as
an SR) can request any known **fixes to problems for a list** of supported
components from an SP (or SP/R acting as an SP). The SP sends any
fixes to
problems associated with the list of supported components to the
requesting SR.

Detailed Description Text - DETX (47):
   Final or **fix FRU list,** in order of decreasing probability

Detailed Description Text - DETX (114):
   FIGS. 8-12 show how problems are detected, determined, and reported
either
by a service requestor or for a service requestor remotely by a service
provider. These flowcharts are executed by processors 111, and 131 of
SR 110,
and SP/R 130, (FIG. 1A) by elements Resource Manager 220, UPPR utility
247, PAR
utility 244, SSF 248, RAS utilities 240, RAS Manager 241, and PDPs 246
(FIGS.
2A and 2B). For purposes of this discussion, SP 150 is considered to
have the
elements of SP/R 130 as shown in FIG. 2B if **remote** problem detection
and
determination is to be performed.

Detailed Description Text - DETX (125):
   Block 820 determines whether a SP/R wants to perform **remote** problem
detection and determination on an SR that it supports. If so, an
operator at
the SP/R console (which is a special one of the terminals 133 or 153 of
FIG. 1A
reserved for network operators) is connected remotely with the SR and
is
permitted to sign on to the SR computer system. Of course, the
operator must
have been given a user id and password to enable access on the SR
system. Once
connected, however, the operator at the SP can initiate the subroutines
shown
in FIGS. 11 and 12 to perform **remote** problem detection and
determination.

Detailed Description Text - DETX (128):
   In the preferred embodiment, the session required to perform **remote**
problem
detection and determination is an APPC session (LU 6.2) in an APPN
network,
although other types of known connections, such as leased, switched or
public
data network, could also be used.

Detailed Description Text - DETX (153):

As has been discussed, a SR can perform problem prevention on itself by
requesting any known **fixes to problems for a list** of supported
components. As
FIG. 8 shows, block 840 asks if **fixes are requested for supported
programs**. If
so, subroutine 1400 of FIG. 14 is called. Block 1401 defines the type
of
problem prevention request desired. Problem prevention can be done at
time of
enrollment, where the SR desires to receive all fixes for all
components it is
requesting the SP to support. Problem prevention could also be done
periodically for a particular component. For example, Pete's Catering
may
decide that it wants to be continually updated on changes to its
Spreadsheet
program. Therefore, on the first of every month, a request is
generated
automatically by Pete's system for problem prevention for the
spreadsheet
program. Problem prevention could also be done at the request of an
operator
at an SR for one or more selected components. The information needed
to
determine the type of problem prevention request desired is stored in
support
database 203.

US-PAT-NO: 4796206

DOCUMENT-IDENTIFIER: US 4796206 A

TITLE: Computer assisted vehicle service featuring
signature

                        analysis and artificial intelligence

       ---------- KWIC ---------

Detailed Description Text - DETX (28):
    Fault Isolation **Repair Procedure Program** Executor (FEX) Diagnostic
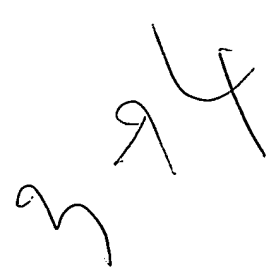Executive

Detailed Description Text - DETX (29):
    Remote Update Facility

Detailed Description Text - DETX (47):
    The Fault Isolation **Repair Procedure Program** Executor (FEX) is the
systems
control program that operates the Technician Terminal. The Vehicle
Service
Procedures operate under control of FEX to perform the following
functions:

Detailed Description Text - DETX (74):
    In the Fault Sequencing Procedure 20 (FSP), for each fault code
passed by
the FAP, there is a list in the form of tables 21 containing the
name(s) of one
or more isolation procedures. These lists are also updated by the host
system
to reflect the results of statistical analysis of service histories.
Each
procedure name in the list represents a component that has required a
repair
action when the related fault code was **detected**. If the list contains
more
than one name, they are ordered based upon frequency of cause.

US-PAT-NO: 6442605

DOCUMENT-IDENTIFIER: US 6442605 B1

TITLE: Method and apparatus for system maintenance on
an image
                        in a distributed data processing system

---------- KWIC ---------

Brief Summary Text - BSTX (14):
   With managed PCs in a network computing environment, the client's
disk image
is a **remote** image from a server. Installing applications at these PCs
are a
difficult and time consuming task. The difficulty lies partly in the
fact that
the user of the PC is typically not well versed in the data processing
system
and application dependencies. Further, most users are not well
prepared to
administer and support installation problems. The installation and
support of
applications usually entails a large degree of expertise and time. As
a
result, an IS or other network manager is usually in charge of
installing and
maintaining applications on PCs in a network computing environment.
Performing
application maintenance on PCs in a network requires the network
manager to
travel to each PC requiring maintenance. When a patch to **fix a problem
in a**
program is applicable to all PCs in a network, the network manager must
travel
to each PC to install the patch and test the system to see if the patch
fixes
the problem. This type of maintenance is time consuming, especially
when in
many cases PCs requiring maintenance may be located on different
floors,
different buildings, or even in different cities.
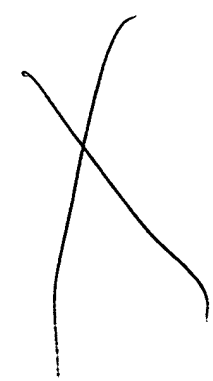
US-PAT-NO: 5784549

DOCUMENT-IDENTIFIER: US 5784549 A

TITLE: Reduced or fail-safe bootstrapping of a system having a

                         graphical user interface

---------- KWIC ---------

Abstract Text - ABTX (1):
    A method and system are disclosed for invoking a limited or
"fail-safe"
functionality from a computer system having a graphical user interface
(GUI).
In one aspect, a computer operating system having GUI functionality can
be
activated with two different sets of GUI features. Normally, the
operating
system provides a full complement of GUI features. Alternatively, the
operating system provides a more limited set of GUI features. The
operating
system invokes the limited set of GUI features either in response to a
user
command, such as a keyboard function key pressed during the bootstrap
loading
sequence of the operating system, or automatically in response to a
failure of
an attempt by the computer system to load the normal complement of GUI
functions. With the limited set of GUI features, the user of the
computer can
take advantage of the power of graphical user interfaces even when the
computer
system is not operating normally. For example, the user can run
graphically
based diagnostic and **repair programs from the "fail**-safe" mode. The
invention
finds application in stand-alone and networked computer systems and, in
particular, in systems that store the bulk of the operating system
software on
a **remote** server accessible only by network.

---------- KWIC ---------

Detailed Description Text - DETX (4):
    FIG. 1 and the following discussion are intended to provide a brief,
general
description of a suitable computing environment in which the invention
may be
implemented. While the invention will be described in the general
context of
an application program that runs on an operating system in conjunction
with a
personal computer, those skilled in the art will recognize that the
invention
also may be implemented in combination with other program modules.
Generally,
program modules include routines, programs, components, data
structures, etc.
that perform particular tasks or implement particular abstract data
types.
Moreover, those skilled in the art will appreciate that the invention
may be
practiced with other computer system configurations, including
hand-held
devices, multiprocessor systems, microprocessor-based or programmable
consumer
electronics, minicomputers, mainframe computers, and the like. The
invention
may also be practiced in distributed computing environments where tasks
are
performed by **remote** processing devices that are linked through a
communications
network. In a distributed computing environment, program modules may
be
located in both local and **remote** memory storage devices.

Detailed Description Text - DETX (7):
    The personal computer 20 may operate in a networked environment
using
logical connections to one or more **remote** computers, such as a **remote**
computer
49. The **remote** computer 49 may be a server, a router, a peer device or
other
common network node, and typically includes many or all of the elements
described relative to the personal computer 20, although only a memory
storage

device 50 has been illustrated in FIG. 1. The logical connections depicted in
FIG. 1 include a local area network (LAN) 51 and a wide area network (WAN) 52.
Such networking environments are commonplace in offices, enterprise-wide
computer networks, intranets and the Internet.

Detailed Description Text - DETX (8):
   When used in a LAN networking environment, the personal computer 20 is
connected to the LAN 51 through a network interface 53. When used in a WAN
networking environment, the personal computer 20 typically includes a modem 54
or other means for establishing communications over the WAN 52, such as the
Internet. The modem 54, which may be internal or external, is connected to the
system bus 23 via the serial port interface 46. In a networked environment,
program modules depicted relative to the personal computer 20, or portions
thereof, may be stored in the **remote** memory storage device 50. It will be
appreciated that the network connections shown are exemplary and other means of
establishing a communications link between the computers may be used.

Detailed Description Text - DETX (15):
   To begin the example, a user initiates the installation of the product, such
as the Microsoft Word application program. Initiating the installation may
constitute the act of inserting CD-ROM disk 31 in optical disk drive 30, or the
act of executing a **remote** startup script over a network connection, or any
other means of initiating the installation. Once the installation is
initiated, the installer application 201 assumes control of processing.

Detailed Description Text - DETX (25):
   On occasion, after a product is installed there arises a need to revise,
through a patch, certain resources of the product, typically to **fix a
programming bug or error**. FIG. 3 is a functional block diagram illustrating
how, in accordance with the invention, a patch is applied to an installed
product, such as the Microsoft Word application program installed in the
example illustrated by FIG. 2. FIG. 4 is a logical flow diagram depicting the
steps performed by the installer application 201 to apply the patch. FIG. 4,

taken in conjunction with FIG. 3, illustrate the steps performed by the installer application 201 to apply a patch to an installed product.

Current US Original Classification - CCOR (1):
   717/124

Current US Cross Reference Classification - CCXR (1):
   717/174

US-PAT-NO: 6151709

DOCUMENT-IDENTIFIER: US 6151709 A

TITLE: Processes and apparatuses for uploading
instructions to
                         a computer

---------- KWIC ---------

Brief Summary Text - BSTX (6):
    As the complexity and pervasiveness of computer systems and their
software
has grown, so too has the need for their technical support.
Troubleshooting of
computer system and software problems generally requires the **remote**
execution
of computer programs or instructions on a customer's computer to
diagnose or
locate the source of the problem. For example, the instructions might
be used
to locate and correct the source of corruption in a database. While
the use of
computer programs to diagnose system and software problems is common
place,
there are many obstacles to efficient implementation of these
diagnostic
programs. For example, the diagnostic program must be transferred to
the
remote computer and sometimes may need to be customized thereat in
order to
address the specific problems at the **remote** computer. In addition,
operation
of the diagnostic program and interpretation of any output therefrom
may
require a level of skill in the art greater than at the customer's
disposal.
Still further, security concerns and the protection of trade secret
information
may necessitate deletion of the diagnostic program after it is no
longer
needed.

Brief Summary Text - BSTX (7):
    Often times, the above-described considerations necessitate a
customer
support representative or developer traveling to the **remote** location of
the
computer system or software problem to load and execute the diagnostic
computer
program. This procedure, however, is often expensive, inefficient and
time
consuming. As such, there is a continuing need for apparatuses and

processes
for providing efficient and cost effective **remote** diagnosis and repair
of
computer system and software problems. Still further, there is a
continuing
need to provide processes and apparatuses for the intelligent
monitoring and
product support of a computer system, wherein a computer application
would be
automatically monitored and upgraded as needed.

Brief Summary Text - BSTX (9):
   Accordingly, it is an object of the present invention to obviate the
above-described shortcomings of **remote** diagnosis of computer systems
and
software.

Brief Summary Text - BSTX (10):
   It is another object of the present invention to provide processes
and
apparatuses for uploading instructions to a **remote** computer for dynamic
execution thereat.

Brief Summary Text - BSTX (11):
   It is still another object of the present invention to provide
processes and
apparatuses for uploading a set of instructions into the working memory
of a
remote computer for dynamic execution by a computer application.

Detailed Description Text - DETX (2):
   Reference will now be made in detail to the preferred embodiments of
the
present invention, examples of which are illustrated in the
accompanying
drawings wherein like numerals indicate the same elements throughout
the views,
and wherein numerals having the same last two digits (e.g., 20, 120)
connote
corresponding structures between the embodiments. As will be
understood
hereafter, the preferred embodiments of the present invention relate to
processes and apparatuses for uploading a set of instructions into the
working
memory of a **remote** computer for execution thereat, wherein the set of
instructions comprises computer code which replaces or supplements the
original
computer code for one or more functions of an existing computer
application
disposed at the **remote** computer. More preferably, execution of the
replacement
or supplemental computer code is initiated by the computer application
when the
computer application is already in a partially executed state (i.e.,

after the
computer application is resident or loaded at the **remote** computer, such as
would be the case with a TSR program). Most preferably, the present invention
is adapted to upload a set of instructions from a client computer to a server
computer to facilitate the diagnosis and/or repair of computer system or
software problems at the server computer and the set of instructions is further
adapted to send information to the client computer to aid in the diagnosis,
repair or updating process.

Detailed Description Text - DETX (13):
   Referring to FIGS. 2, 4 and 5 and in accordance with another aspect of the
present invention, an exemplary process 82 for uploading and executing the set
of instructions 48 at the second computer 24 will now be described in the
context of diagnosing or **fixing a problem with the server application** 54
disposed at the second computer 24. A user of the second computer 24, such as
a system administrator, upon encountering a problem (such as data corruption, a
conflict, or other reportable error) with the server application 54 contacts
the user of the first computer 22, who may be, for example, the developer of or
a technical support person for the server application 54, to advise this person
of the problem. The user of the second computer 24 then re-initiates execution
of the server application 54 and, through the use of a switch or other input,
triggers execution of the switch routine 72 to ready the server application 54
for the uploading of the set of instructions 48 into the second computer's RAM
28. The switch routine 72 preferably disables any server functions 68 or other
processes (e.g., threads, tasks, etc.) of the server application 54 which could
interfere with execution of the set of instructions 48. The switch routine 72
also preferably spawns or initiates a separate thread or task for monitoring
the communication port of the second computer 24 for a packet 40 from the first
computer 22 for establishing a network connection there between, as shown in
block 84 of FIG. 4.

Current US Original Classification - CCOR (1):
717/173